

TITLE OF THE INVENTION

Document Structure Inspection Method and Apparatus

The present application claims priority from the
5 Japanese patent application JP2003-140568 filed on May 19,
2003, the content of which is hereby incorporated by reference
into this application.

BACKGROUND OF THE INVENTION

10 The present invention relates to a structure inspection
method for a structured document.

In a computer system where a process request is received
via a communication network to perform a requested application
process, the data structure of an electronic document
15 describing the request is generally checked for consistency
with a predetermined structure each time it is received. The
computer used in the system performs the requested application
process and returns the result only when the document structure
is consistent with the predetermined structure. If, on the
20 other hand, the document structure is inconsistent with the
predetermined structure, the computer returns an error to the
requesting source.

The above document structure inspection process can be
performed simultaneously with an application process.

However, if any document structure inconsistency is found at a final stage of the application process, all the previously performed application process steps are wasted so that the processing efficiency decreases. To prevent the processing efficiency from decreasing, it is preferred that the entire document structure be inspected before the application process. This also holds true when the received electronic document is expressed in XML (Extensible Markup Language).

The interface of an inspection process for checking whether the structures of various structured documents are consistent with a predefined document structure is stipulated as described under "Schema Validation" in "Java API for XML Processing Version 1.2 Final Release" (Reference Document 1), which is written by Rajiv Mordani et al. with reference to Java (registered trademark of Sun Microsystems). Further, "Xerces 2 Java Parser Readme" (Reference Document 2) of the Apache Software Foundation describes how to implement a structure inspection process for such structured documents.

In a computer system for transmitting/receiving electronic documents via a communication network, the electronic documents to be transmitted may be electronically signed by a sender or encrypted for the purpose of preventing the transmitted documents from being falsified or viewed by an unauthorized person.

In a computer system comprising a plurality of computers, which sequentially perform various processes to carry out a complicated process as a whole, a process requesting person, who is the initial sender, may electronically sign or encrypt
5 part of the data to be transmitted.

The above processing operation is performed, for instance, for on-line shopping. When a payment is made with a credit card for on-line shopping purposes, the data to be transmitted from the purchaser (initial sender) to an on-line
10 shopping site manager includes the information identifying the item to be purchased and the information identifying the credit card (card number, expiration date, etc.) used for payment. The information about the credit card is forwarded from the on-line shopping site manager to a credit card company where
15 a credit-card settlement process is performed. The on-line shopping site manager does not directly use the credit card information, but receive it from the purchaser and relay it to the credit card company.

In the system described above, a higher degree of safety
20 is achieved by the use of a transmission method in which the purchaser encrypts part of the credit card information so as to permit only the credit card company to decrypt it while keeping it from being disclosed to the on-line shopping site manager than the use of a transmission method in which the data

to be transmitted is entirely encrypted by the on-line shopping site manager. In this instance, the encrypted data may be electronically signed (partial signature) to indicate that the credit card information is encrypted by the purchaser. It is anticipated that the above electronic transactions will prevail in the future due to widespread use of a public key cryptography technology.

There is a technology that converts a document structure definition written in a certain document structure definition language into an equivalent document structure definition written in another language. For example, a conversion tool named "dtd2xs" is introduced in "LuMrix, 'XML Tools', 'online', 2003, [retrieved on February 14, 2003], Internet <URL: <http://puvogel.informatik.med.uni-giessen.de/lumrix/#dtd>>" (Reference Document 3). This tool converts a document structure definition written in document definition language DTD into an equivalent document structure definition written in "W3C XML Schema". When used to effect conversion among various document structure definition languages, however, the tool does not yield a perfectly equivalent document structure definition due, for instance, to different description capabilities of the document structure definition languages.

SUMMARY OF THE INVENTION

When the structured document to be transmitted/received is signed, encrypted, or otherwise processed using a conventional technology as described above, the document structure changes. Therefore, the same structure inspection method cannot be used as for documents that are not signed, encrypted, or otherwise processed. If the structured document to be transmitted/received is merely signed, a structure inspection can be conducted after the entire signature is removed. However, the structure inspection cannot be conducted if the on-line shopping site manager receives encrypted credit card information that the on-line shopping site manager cannot decrypt, as described earlier.

It is an object of the present invention to make it possible to entirely inspect the structure of a structured document even when it is changed by signing or encrypting the structured document wholly or partly or when an encrypted portion cannot be decrypted by an inspector.

To handle a structured document whose structure may be changed, for instance, by electronically signing an electronic structured document or by encrypting part of such a document and achieve the above object, the present invention sets up a document structure alteration rule, converts the document structure definition into a signed/encrypted document structure definition before inspecting the document structure

definition, which expresses the structured document structure, and conducts a document structure inspection of the converted document structure definition.

5 BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates a computer environment;

Fig. 2 illustrates the internal hardware configuration of a computer;

Fig. 3 shows an example of a document structure
10 definition;

Fig. 4 shows an example of a structured document;

Fig. 5 shows an example of a document structure alteration rule library according to a first embodiment of the present invention;

15 Fig. 6 shows an example of a document structure definition;

Fig. 7 shows an example of a document structure definition;

Fig. 8 shows an example of a document structure
20 definition;

Fig. 9 shows an example of a document structure definition;

Fig. 10 is a flowchart illustrating the processing steps according to the first embodiment of the present invention;

Fig. 11 shows an example of a document structure definition to which document structure alteration rule 1 has been applied;

Fig. 12 shows an example of a document structure definition to which document structure alteration rules 1 and 2 have been applied;

Fig. 13 shows an example of a document structure definition to which document structure alteration rules 1, 2, and 3 have been applied;

Fig. 14 shows an example of a structured document whose structure has been altered;

Fig. 15 shows an example of a document structure alteration rule;

Fig. 16 shows an example of a document structure definition to which document structure alteration rule 4 has been applied;

Fig. 17 illustrates the internal hardware configuration of a computer according to a second embodiment of the present invention;

Fig. 18 shows an example of a document structure alteration rule library according to the second embodiment of the present invention;

Fig. 19 is a flowchart illustrating in detail the processing steps for document structure inspection;

Fig. 20 illustrates the status of a stack;

Fig. 21 illustrates the status of a stack; and

Fig. 22 shows an example of a document structure definition that is formulated by means of referencing.

5

DETAILED DESCRIPTION OF THE INVENTION

A document structure inspection method and apparatus that can cope with signature- or encryption-induced structural changes in a structured document will be described below. The
10 present invention can also cope with document structure changes that are applied by an operation other than a signature or encryption.

A first embodiment of the present invention will now be described with reference to the accompanying drawings. As
15 shown in Fig. 1, the present invention relates to an inspection method and system in which a document recipient checks whether the data structure of a received document is consistent with a predetermined one in an environment where a plurality of computers 101, 102, 104 are interconnected via a hard-wired
20 or wireless network 103 to mutually exchange information.

The description of the present embodiment assumes that information is exchanged between two parties. However, the present invention can also be applied to the communication between two arbitrary parties in a situation where three or

more parties are involved in information exchange. More specifically, when the communication involves three parties so that sender A transmits information to recipient C via recipient B, the present invention is applicable to two
5 two-party communications, that is, the information transmission from sender A to recipient B and the information transmission from sender B to recipient C.

Fig. 2 illustrates the configuration of a recipient's computer. The recipient's computer 104 comprises a CPU 201,
10 a memory 202, a display device 203, an input device 204, a communications device 205, and a storage device 206, and is connected to the network 103. The storage device 206 stores a document structure inspection unit 211, an application processor 212, a document structure definition library 213,
15 a document structure definition converter 214, and a document structure alteration rule library 215. The document structure inspection unit 211 is a program for inspecting the structure of a received document. The application processor 212 is a program for processing an application request included in the
20 received document. The document structure definition library 213 stores a document structure definition that describes the structure of the received document. The document structure alteration rule library 215 stores document structure alteration rules, which state how the document structure is .

altered by an digital signature, encryption, or other operation performed on the outgoing document. The document structure definition converter 214 is a program for converting the document structure definition in compliance with a
5 document structure alteration rule.

The document structure inspection unit 211, application processor 212, and document structure definition converter 214, which are stored in the storage device 206, are loaded into the memory and executed by the CPU. The present invention is
10 implemented in this manner.

The document structure inspection unit 211 according to the first embodiment is the same as disclosed in Reference Document 2. More specifically, the document structure inspection unit 211 conducts an inspection on an individual
15 element name basis to determine whether a received structured document is consistent with the corresponding document structure definition. If the inspection reveals that the received document is consistent with the definition, the application processor 212 performs an application process in
20 compliance with a request specified by the received document. Prior to document structure inspection, the document structure definition converter 214 performs a document structure definition conversion process on the received document in compliance with a document structure alteration rule. After

completion of the conversion process, the document structure inspection unit 211 conducts an inspection in accordance with the document structure definition.

Fig. 3 shows an example of a document structure definition that is contained in the document structure definition library 213. This document structure definition is written in DTD (Document Type Definition). DTD is a document structure definition language that is designed to define the structure of XML and other documents. Although the definition is written in DTD in this example, the document structure may be defined in some other document structure definition language. An XML document is an aggregate of elements of a nested structure. The leading element is called a "root element". When element E is included in element F, the relationship between elements E and F is called a "parent-child relationship". Element E is referred to as a "parent element" for element F. Element F is referred to as a "child element" for element E. Elements having the same parent element are called "sibling elements". Each element can have zero or more attributes.

For explanation purposes, a 2-digit line number is attached to the beginning each line of document structure definition in Fig. 3. However, the real document structure definition does not contain any line number. Similarly, the

structured documents and document structure definitions referenced for subsequent explanations are provided with line numbers. However, such line numbers are not included in the actual structured documents or document structure

5 definitions.

The first line (line No. 1) in Fig. 3 indicates that the root element is "PurchaseOrder". The second line (line No. 2) indicates that the "PurchaseOrder" element has three child elements ("UserID", "Price", and "CreditCard"), and that
10 these child elements appear only once in the order named. The third line (line No. 3) indicates that the "PurchaseOrder" element has an "Id" attribute, and that if the attribute exists, its value must be unique within the XML document, and further that the attribute does not always have to exist. The fourth
15 line (line No. 4) and fifth line (line No. 5) respectively indicate that arbitrary character strings are acceptable as the "UserID" and "Price" element values. The sixth line (line No. 6) indicates that the "CreditCard" element has four child elements ("Issuer", "Number", "Expire", and "Owner"), and that
20 these child elements appear once only in the order named. The seventh line (line No. 7), eighth line (line No. 8), ninth line (line No. 9), and tenth line (line No. 10) respectively indicate that arbitrary character strings are acceptable as the "Issuer", "Number", "Expire", and "Owner" element values. It

is assumed that the above document structure definition is named "PurchaseOrder.dtd" and stored in the document structure definition library 213.

Fig. 4 shows an example of an XML document that is
5 consistent with the document structure definition
"PurchaseOrder.dtd". For ease of explanation, however, the name space, which is contained in a normal XML document, is omitted from the example. The name space is used to maintain element name uniqueness within a single XML document. In the
10 example that is formulated to assure element name uniqueness without the name space, no problem arises even when the name space is excluded.

The first line (line No. 1) in Fig. 4 indicates that this document is an XML document. The second line (line No.
15 2) indicates that the structure of the XML document is consistent with the definition "PurchaseOrder.dtd". The third line (line No. 3) and subsequent lines indicate the contents of the documents. The fourth line (line No. 4) indicates that the UserID element value is "10194970". The
20 fifth line (line No. 5) indicates that the Price element value is "100000".

Fig. 5 shows some typical document structure alteration rules that are contained in the document structure alteration rule library 215. Three rules (rules 511, 512, and 513) are

indicated in the figure. Each rule comprises a type 501, which indicates a conversion method, an applied location 502, which uses an element name to indicate the document structure definition targeted for conversion and the applied spot, an operation element 503, which results from conversion, relevant document structure definitions 504, which are document structure definition sources prevailing after conversion, and a document structure alteration rule number 505.

One of three different types ("Replace", "Add", or "Delete") is indicated as the type 501. However, the "Delete" type is not shown in Fig. 5. "Replace" means that the element indicated as the applied location 502 is to be replaced by the element indicated as the operation element 503. "Add" means that the element indicated as the operation element 503 is to be added to the location indicated as the applied location 502. "Delete" means that the element indicated as the applied location 502 is to be deleted. If the type is "Delete", the fields for the operation element 503 and relevant document structure definitions 504 may be left "blank".

The applied location 502 is expressed by a character string that comprises a character string indicating a document structure definition and a character string indicating a location within the definition. These two character strings are joined with a colon (:). The location within the document

structure definition is expressed by a character string that is formulated by sequentially joining the tree structure root and subordinate element names including the specified element name with a slash (/).

5 A specific location within the tree structure may be expressed, instead of specifying a certain element, by using "first()", which indicates a location prior to the leading child element of a certain element, "last()", which indicates a location after the trailing child element of a certain element, 10 "before()", which indicates a location immediately before a certain element (between a certain element E and a sibling element D if sibling element D is located immediately before element E), or "after()", which indicates a location immediately after a certain element (between a certain element 15 E and a sibling element F if sibling element F is located immediately after element E). Further, "value()", which represents a certain element value, may be used to express a specific location. It is also possible to express a range by joining contiguous sibling elements with the symbol "-". For 20 example, "po.dtd:/po/value()" denotes the value of the root element "po" for the document structure definition "po.dtd". On the other hand, "po.dtd:/po/one-three" denotes a range from the child element "one" to the child element "three" where the child elements "one" to "three" are contiguous child elements

of the root element "po" for the document structure definition "po.dtd".

For example,

"PurchaseOrder.dtd:/PurchaseOrder/CreditCard" under
5 "Applied location" 502 for the first document structure alteration rule 511 indicates the child element "CreditCard", which belongs to the root element "PurchaseOrder" for the document structure definition "PurchaseOrder.dtd".
"PurchaseOrder.dtd:/PurchaseOrder/last()" under "Applied
10 location" 502 for the second document structure alteration rule 512 indicates a location after the trailing child element, which belongs to the root element "PurchaseOrder" for the document structure definition "PurchaseOrder.dtd". Although the above special notation is used in the example, any notation
15 can be used as far as it can indicate a specific document structure definition location.

The field for "Operation element" 503 uses the same notation as the field for "Applied location" 502 to express an element for replacement or addition. For example, the
20 operation element "EncryptedData.dtd:/EncryptedData" for the first document structure alteration rule 511 indicates the root element "EncryptedData" for the document structure definition "EncryptedData.dtd".

The field for the relevant document structure definitions 504 indicates document structure definitions required for document structure conversion. This field may contain a plurality of document structure definitions. For example, the relevant document structure definition entry "EncryptedData.dtd KeyInfo.dtd" for the first document structure alteration rule 511 indicates that two document structure definitions ("EncryptedData.dtd" and "KeyInfo.dtd") relate to the first document structure alteration rule. All the document structure definitions appearing in the document structure alteration rule library 215, including the document structure definitions indicated in the field for the relevant document structure definitions 504, must be acquired beforehand and stored in the document structure definition library 213.

Fig. 6 shows an example of EncryptedData.dtd. Fig. 7 shows an example of EncryptedKey.dtd. Fig. 8 shows an example of Signature.dtd. Fig. 9 shows an example of KeyInfo.dtd. For explanation purposes, the contents of these examples are simplified in compliance with the XML document encryption specification (XML Encryption) and XML document signature specification (XML Signature), which are drawn up by the W3C (World Wide Web Consortium).

Fig. 10 is a flowchart illustrating the processing steps that are performed by the document structure definition converter 214. The document structure definition converter 214 first judges whether all the document structure alteration rules have been applied to the received document structure definition describing the document structure (step 1001). If all the alteration rules have been applied, the document structure definition converter 214 terminates its process. If any alteration rule is unapplied, the document structure definition converter 214 acquires the next alteration rule that matches the applied location 502 (step 1002). When no more alteration rules are left unapplied, the document structure definition converter 214 terminates its process in accordance with the judgment formulated in step 1001. Next, the document structure definition converter 214 examines the type 501 of an acquired alteration rule. If the type is "Replace" (step 1003), the element written at the alteration rule's applied location 502 within the targeted document structure definition is replaced by the operation element 503 (step 1010). If the type is "Add" (step 1004), the operation element 503 is added to the alteration rule's applied location 502 within the targeted document structure definition (step 1011). After the replacement process (step 1010) and addition process (step 1011), the document structure definition

converter 214 also adds to the document structure definition library 213 the replacement element, added element, and the element structure definitions used for defining the replacement element and added element (step 1012).

5 Application examples of the three document structure alteration rules 511, 512, 513 in Fig. 5 will now be described.

 The first document structure alteration rule 511 is applied to the document structure definition "PurchaseOrder.dtd", which is shown in Fig. 3. Fig. 11 shows
10 the resulting state. Since the type of the first document structure alteration rule 511 is "Replace" while the applied location is "PurchaseOrder.dtd:/PurchaseOrder/CreditCard" with the operation element set to "EncryptedData.dtd:/EncryptedData", the child element
15 "CreditCard" (320 in Fig. 3) under the root element "PurchaseOrder" for "PurchaseOrder.dtd" is replaced by the root element "EncryptedData" (1121) for "EncryptedData.dtd". It is defined that the unconverted root element "PurchaseOrder" has three child elements, which are "UserID",
20 "Price", and "CreditCard" (line No. 02 in Fig. 3). After conversion, the root element "PurchaseOrder" has three child elements, which are "UserID", "Price", and "EncryptedData" (line No. 02 in Fig. 11).

 Lines No. 11 through 16 (1114) and 17 through 22 (1115)

in Fig. 11 add the two document structure definitions "EncryptedData.dtd" and "KeyInfo.dtd", which are enumerated as the relevant document structure definitions 504 for the first document structure alteration rule 511, to the document structure definition "PurchaseOrder.dtd", which is targeted for application. In this example, the entities of the document structure definitions are added. Alternatively, however, only the references to the corresponding document structure definitions (information for identifying the document structure definitions) may be added with the entities stored in the document structure definition library 213 separately from "PurchaseOrder.dtd". The document structure definition 2201 shown in Fig. 22 is an example in which the document structure definition 1101 shown in Fig. 11 is expressed by using the references to the document structure definitions "EncryptedData.dtd" and "KeyInfo.dtd". Lines No. 11 and 12 (2211) in Fig. 22 use references to express the portions corresponding to lines No. 11 through 16 (1114) in Fig. 11. Lines No. 13 and 14 (2212) in Fig. 22 use references to express the portions corresponding to lines No. 17 through 22 (1115) in Fig. 11. Fig. 11 shows an example in which some of the document structure definition elements in a structured document are replaced by document structure definition elements corresponding to encrypted elements.

Since the "CreditCard" element is rendered unnecessary by the element replacement process, its definition (line No. 06) and the child elements for "CreditCard" (lines No. 07 through 10 (1113) in Fig. 11) may be deleted from the document structure definition "PurchaseOrder.dtd".

Fig. 12 shows the document structure definition "PurchaseOrder.dtd" to which the second document structure alteration rule 512 has been applied. Since the type of the second document structure alteration rule 512 is "Add" while the applied location is "PurchaseOrder.dtd:/PurchaseOrder/last()" with the operation element set to "EncryptedKey.dtd:/EncryptedKey", the root element "EncryptedKey" for "EncryptedKey.dtd" is added as the last element for the root element "PurchaseOrder" of "PurchaseOrder.dtd" (1211 of line No. 02 in Fig. 12). Further, the document structure definition "EncryptedKey.dtd", which is enumerated as a relevant document structure definition for the second document structure alteration rule, is added to the document structure definition "PurchaseOrder.dtd", which is targeted for application (lines No. 23 through 29 (1212) in Fig. 12).

Finally, Fig. 13 shows the document structure definition "PurchaseOrder.dtd" to which the third document structure alteration rule 513 has been applied. In the example

shown in Fig. 13, the root element "Signature" for "Signature.dtd" is added as the last element for the "PurchaseOrder" element (1311 of line No. 02) with the structure definition (1312 of lines No. 30 through 41) for the "Signature" element added. The element added in Fig. 13 is an example of an element corresponding to an digital signature affixed to a structured document.

Now that all the document structure alteration rules in the document structure alteration rule library 215 have been applied, the document structure definition converter 214 terminates its process.

When the above document structure conversion process is completed, it is possible to conduct a structure inspection of a structured document shown in Fig. 4 after it is encrypted and electronically signed (Fig. 14). Lines No. 06 through 15 (1411) in Fig. 14 show an encrypted "CreditCard" element. Lines No. 16 through 27 (1412) show key information used for encryption. Lines No. 28 through 41 (1413) show a signature.

The above example merely shows a document structure alteration rule for changing the "PurchaseOrder.dtd" structure. Alternatively, however, it is possible to define a document structure alteration rule that changes the structure of a document structure definition other than "PurchaseOrder.dtd" shown in Fig. 15.

Reference numeral 1511 in Fig. 15 shows an digital signature method. It is an example of a document structure alteration rule that corresponds to a method for causing a signature to include a document targeted for signature (this method is generally referred to as a "enveloping signature").

Fig. 16 shows an example of document structure definition conversion for a situation where the above signature method is applied. Since the type of document structure alteration rule 1511 is "Add" while the applied location is "Signature.dtd:/Signature/last()" with the operation element set to "PurchaseOrder.dtd:/PurchaseOrder", the root element "PurchaseOrder" for the document structure definition "PurchaseOrder.dtd" is added to line No. 02 as the last child element for the root element "Signature" of the document structure definition "Signature.dtd" (1611). Further, "PurchaseOrder.dtd" and "KeyInfo.dtd", which are enumerated as the relevant document structure definitions, are added to "Signature.dtd" as lines No. 14 through 22 (1612) and 23 through 28 (1613). When this document structure alteration rule is applied, the root element for the received document is changed to "Signature".

As described above, the first embodiment of the present invention makes it possible to conduct a structure inspection of the entire structured document even when the document

structure is altered by signing or encrypting the structured document wholly or partly or when encrypted portions cannot be decrypted by an inspector.

A second embodiment of the present invention will now be described with reference to the accompanying drawings. Fig. 17 illustrates the hardware configuration of the second embodiment according to the present invention. The hardware configuration of the second embodiment according to the present invention is basically the same as that of the first embodiment. The differences lie in the structure inspection processing steps, which are performed by a document structure inspection unit 1701 within the storage device 206, and in the data structure of a document structure alteration rule library 1711.

Fig. 18 shows some typical document structure alteration rules that are contained in the document structure alteration rule library 1711. Four rules (rules 1811, 1812, 1813, and 1814) are indicated in the figure. Each rule comprises a type 1801, which indicates a conversion method, an applied definition 1802, which indicates an applied document structure definition, an operation element 1803, which is targeted for conversion, relevant document structure definitions 1804, which are required for document structure

inspection after conversion, and a document structure alteration rule number 1805.

Either "Replace" or "Add" is indicated as the type 1801. "Replace" means that part of the document structure definition indicated as the applied definition 1802 may be replaced by the document structure definition element indicated as the operation element 1803. "Add" means that the document structure definition element indicated as the operation element 1803 may be added to part of the document structure definition indicated as the applied definition 1802. In the field for the applied definition 1802, either a document structure definition name or the symbol "*" is written. The symbol "*" means that any document structure definition is acceptable as the applied definition. The notations for the operation element 1803 and relevant document structure definitions 1804 conform to those for the operation element 503 and relevant document structure definitions 504 that are shown in Fig. 5 and described in conjunction with the first embodiment.

Fig. 19 is a flowchart illustrating the processing steps that are performed by the document structure inspection unit 1701. The document structure inspection unit 1701 first identifies a document structure definition that corresponds to a structured document targeted for inspection, and sets it

as the current document structure definition (step 1901). The document structure inspection unit 1701 places the inspection positions for both the structured document targeted for inspection and the document structure definition at the beginning. Next, the document structure inspection unit 1701 conducts an inspection by sequentially shifting the inspection position to determine whether the structured document targeted for inspection is consistent with the current document structure definition (step 1902). If the inspection reaches the end of the current document structure definition or the end of the structured document targeted for inspection or if an error occurs, step 1902 ends so that step 1903 is initiated. The "end of the current document structure definition" does not denote the last line of the current document structure definition but indicates the "end of the root element definition in the associated document structure definition".

Step 1903 is performed to determine whether the end of the current document structure definition is reached by the inspection. If the end of the current document structure definition is not reached by the inspection, the cause of an abnormal end is identified. More specifically, step 1904 is performed to check whether the abnormality is caused by the "appearance of an illegal element". If not, the result is returned to indicate an "inconsistency". If the element names

targeted for comparison do not match, the first embodiment concludes that an "inconsistency" is encountered. The second embodiment, on the other hand, concludes that an "illegal element" has appeared. If the abnormality is caused by the

5 "appearance of an illegal element", step 1905 is performed to search the document structure alteration rule library for an applicable document structure alteration rule. The term "applicable" means that the current document structure definition is consistent with the "applied definition" for the

10 document structure alteration rule, and that the illegal element appearing in the structured document targeted for inspection is consistent with the "operation element" for the document structure alteration rule. However, if the symbol "*" is indicated as the "applied definition", it indicates that

15 every document structure definition is complied with.

If no applicable document structure definition is found in step 1905, step 1906 is performed to return the result to indicate an "inconsistency". If an applicable document structure definition is found in step 1905, step 1907 is

20 performed to simultaneously push the name of the current document structure definition, its inconsistent location, and the type of the found document structure alteration rule to the stack, and set the document structure definition of the operation element of the found document structure alteration

rule as the current document structure definition. In the present embodiment, however, the notation used for indicating the inconsistent location is the same as that for indicating the applied location 502 for a document structure alteration rule according to the first embodiment. The document structure inspection unit 1701 then returns to step 1902 and continues with the inspection process.

If the inspection process ends normally in step 1903, step 1908 is performed to check the stack status. If the stack is empty and the end of the structured document targeted for inspection is reached (the question in step 1912 is answered "Yes"), the document structure inspection unit 1701 returns "consistency" and ends normally. If, on the other hand, the end of the structured document targeted for inspection is not reached in step 1912, the document structure inspection unit 1701 returns "inconsistency" and ends.

If the stack is not empty in step 1908, the document structure definition, inconsistent location, and alteration rule type are popped from the beginning of the stack, and the popped document structure definition is set as the current document structure definition (step 1909). Step 1909 is then performed to check whether the popped alteration rule type is "Replace". If the popped alteration rule type is not "Replace", the document structure inspection unit 1701 returns to step

1902 and continues with the inspection process. If, on the other hand, the popped alteration rule type is "Replace", the document structure inspection unit 1701 moves the inspection position for the current document structure definition to the next sibling element. If the next sibling element is not found, step 1911 is performed to move the inspection position to a location immediately after the current element. When the alteration rule popped in step 1909 is not "Replace" or after completion of step 1911, the document structure inspection unit 1701 proceeds to step 1902 and continues with the inspection process.

The document structure inspection process performed as indicated in Fig. 19 to inspect the structured document shown in Fig. 14 will now be described.

It is assumed that the document structure definition library 213 stores "PurchaseOrder.dtd" in Fig. 3, "EncryptedData.dtd" in Fig. 6, "EncryptedKey.dtd" in Fig. 7, "Signature.dtd" in Fig. 8, and "KeyInfo.dtd" in Fig. 9, and that the document structure alteration rule library 1711 stores the four rules shown in Fig. 18.

First, step 1901 is performed to identify the document structure definition as "PurchaseOrder.dtd" from the second line of the structured document targeted for inspection (1401 in Fig. 14). Next, step 1902 is performed to conduct a regular

inspection until an error occurs or the inspection ends. In the currently employed example, the inspection process is performed normally between the third line and fifth line. However, when the element name "EncryptedData" in the sixth line is reached, an error occurs because it is inconsistent with the element name "CreditCard", which is indicated in the second line of the document structure definition (Fig. 3). Since this error is caused by the appearance of the illegal element "EncryptedData", the document structure inspection unit 1701 proceeds to step 1905 and searches the document structure alteration rule library 1711 for a document structure alteration rule in which the applied definition is consistent with the current document structure definition "PurchaseOrder.dtd" and the operation element is consistent with "EncryptedData". As a result of the search, the first document structure alteration rule 1811 shown in Fig. 18 is retrieved. The applied definition for the first document structure alteration rule 1811 is indicated by the symbol "*". Since it is consistent with any document structure alteration rule, it is consistent with "PurchaseOrder.dtd" as well.

Next, step 1907 is performed to push the current document structure definition "PurchaseOrder.dtd", the inconsistent location "/PurchaseOrder/CreditCard", and the type "Replace" for document structure alteration rule 1, which

is found, to the stack, as indicated in Fig. 20.

"PurchaseOrder.dtd|/PurchaseOrder/CreditCard|Replace"

(2001) is stored at the beginning (top) of the stack. Next, the operation element "EncryptedData.dtd:/EncryptedData" for document structure alteration rule 1, which is found in step 1905, is set as the current document structure definition. The document structure inspection unit 1701 then returns to step 1902 and continues with the inspection process.

Next, step 1902 is performed to conduct an inspection of the "EncryptedData" element (line No. 02) for the document structure definition "EncryptedData.dtd" in Fig. 6 and lines No. 06 and beyond of the structured document targeted for inspection 1401 in Fig. 14. When this inspection reaches the end of line No. 15 of the structured document targeted for inspection 1401, the end of the document structure definition 601 is reached. Therefore, the question in step 1903 is answered "Yes" to initiate a stack inspection. Since the stack is not empty as it stores the information 2001 that was pushed in step 1907, the document structure inspection unit 1701 proceeds to step 1909 and pops the information 2001 at the beginning of the stack. The stack is emptied by this process.

Next, the document structure inspection unit 1701 sets the current document structure definition to the previously popped document structure definition "PurchaseOrder.dtd" and

the inspection location to `"/PurchaseOrder/CreditCard"`. The document structure inspection unit 1701 then proceeds to step 1911 because the type of the document structure alteration rule popped in step 1909 is `"Replace"`. Step 1911 is performed so as to select the next sibling element as the inspection position for the current structure definition that was set in step 1909 to `"/PurchaseOrder/CreditCard"`. However, the next sibling element does not exist for the `CreditCard` element. Therefore, the inspection position is set to `"a location immediately after the CreditCard element (/PurchaseOrder/CreditCard/after())"`. The document structure inspection unit 1701 then proceeds to step 1902 and continues with the inspection process.

The inspection position is now set at `"/PurchaseOrder/CreditCard/after()"` for the current document structure definition `"PurchaseOrder.dtd"`, whereas the inspection position for the structured document targeted for inspection (1401 in Fig. 14) is at the end of line No. 15.

When the inspection process is continuously performed in step 1902, the illegal element `"EncryptedKey"` is found to appear in line No. 16 of the structured document targeted for inspection (1401 in Fig. 14). Therefore, the document structure inspection unit 1701 proceeds to step 1905 to search the document structure alteration rule library 1711 for a document structure alteration rule whose operation element is

consistent with the applied definition. Since the current search retrieves the second document structure alteration rule 1812 shown in Fig. 18,

"PurchaseOrder.dtd|/PurchaseOrder/CreditCard/after()|Add"

5 is pushed to the stack. The resulting stack status is shown in Fig. 21.

Further, the document structure inspection unit 1701 sets the current document structure definition to "EncryptedKey.dtd" and the inspection position to the
10 "EncryptedKey" element, and then proceeds to step 1902 to continue with the inspection process.

In step 1902, the end of the current document structure definition "EncryptedKey.dtd" is reached when the inspection reaches the end of line No. 27 of the structured document
15 targeted for inspection 1401 shown in Fig. 14. Therefore, the process proceeds to step 1908. Next, the document structure inspection unit 1701 pops the stack's leading information "PurchaseOrder.dtd|immediately after the CreditCard element in the second line|Add" from the stack, and sets the current
20 document structure definition and the inspection position to "PurchaseOrder.dtd" and "/PurchaseOrder/CreditCard/after()", respectively. Since the currently popped type is "Add", the document structure inspection unit 1701 proceeds to step 1902 and continues with the inspection process.

If the inspection process is continuously performed in step 1902, the illegal element "Signature" appears while the inspection position for the structured document targeted for inspection remains unchanged. Therefore, the document structure inspection unit 1701 proceeds to step 1905 to search the document structure alteration rule library 1711 for a document structure alteration rule whose operation element is consistent with the applied definition. Since the current search retrieves the third document structure alteration rule 1813 shown in Fig. 18,

"PurchaseOrder.dtd|/PurchaseOrder/CreditCard/after()|Add" is pushed to the stack. Further, the document structure inspection unit 1701 sets the current document structure definition to "Signature.dtd" and the inspection position to "Signature", proceeds to step 1902, and continues with the inspection process.

In step 1902, the end of the current document structure definition "Signature.dtd" is reached when the inspection reaches the end of line No. 41 of the structured document targeted for inspection 1401. Therefore, the process proceeds to step 1908. Next, the document structure inspection unit 1701 pops the stack's leading information "PurchaseOrder.dtd|/PurchaseOrder/CreditCard/after()|Add", and sets the current document structure definition and the

inspection position to "PurchaseOrder.dtd" and
"/PurchaseOrder/CreditCard/after()", respectively. The
stack is emptied by this process. Since the currently popped
type is "Add", the document structure inspection unit 1701
5 proceeds to step 1902 and continues with the inspection
process.

In step 1902, the end of the current document structure
definition "PurchaseOrder.dtd" is reached when the inspection
reaches the end of line No. 42 of the structured document
10 targeted for inspection 1401. Therefore, the process proceeds
to step 1908. Since the stack is empty and the end of the
structured document targeted for inspection is reached (the
end of line No. 42 is reached), the document structure
inspection unit 1701 terminates the inspection process and
15 returns the result to indicate a "consistency".

The inspection process is now completed for the
structured document targeted for inspection 1401. The fourth
document structure alteration rule 1814 has not been applied
to the inspection of the structured document targeted for
20 inspection 1401. The fourth document structure alteration
rule is to be used when an inspection is conducted of a
structured document in which a signature includes a document
targeted for signature (a structured document consistent with
the document structure definition shown in Fig. 16).

As described above, the second embodiment of the present invention makes it possible to conduct a structure inspection of the entire structured document even when the document structure is altered by signing or encrypting the structured document wholly or partly or when encrypted portions cannot
5 be decrypted by an inspector.

The first and second embodiments assume that all document structure definitions are written in a single document structure definition language, that is, DTD. However,
10 a structured document targeted for inspection can be subjected to a structure inspection even when document structure definitions are written in a plurality of document structure definition languages. More specifically, such a structure inspection can be conducted by identifying the type of document
15 structure definition language used for writing the document structure definitions and selecting an appropriate document structure inspection unit when the first embodiment acquires the entity of a document structure definition by using a reference to it (2211, 2212) or when the second embodiment
20 performs step 1902. The type of document structure definition language can be determined according to the document structure definition name.

For example, the document structure definition named "PurchaseOrder.dtd" may indicate that it is written in DTD

because the employed extension is "dtd". The document structure definition named "Signature.xsd" may indicate that it is written in "W3C XML Schema" because the employed extension is "xsd".

5 Further, when a plurality of document structure definition languages are used to write document structure definitions, the structure inspection of a structured document targeted for inspection can also be conducted by converting document structure definitions stored in the document
10 structure definition library 213 in such a manner that they are written in a single document structure definition language.

 The present invention makes it possible to conduct a structure inspection of the entire structured document even
15 when the document structure is altered by signing or encrypting the structured document wholly or partly and no matter whether encrypted portions can be decrypted by an inspector.